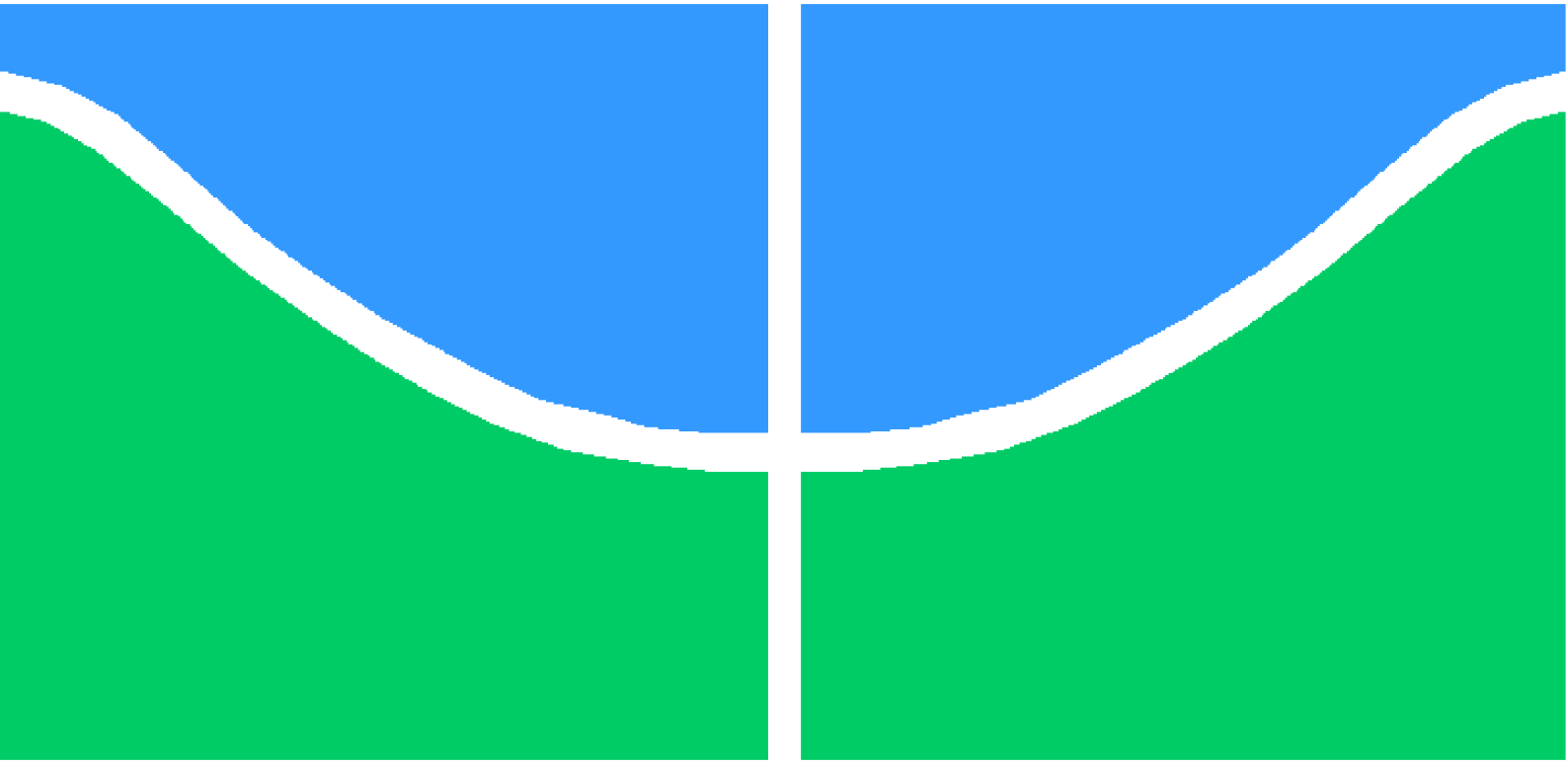

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Busca de Textos Utilizando Similaridade Semântica no Contexto Biológico e Biomédico

Autora: Fabiana Mitsu Alvarenga Ofugi
Orientador: Prof. Dr. Edgard Costa Oliveira
Coorientadora: Prof.ª Dr.ª Milene Serrano

Brasília, DF
2015

Fabiana Mitsu Alvarenga Ofugi

Busca de Textos Utilizando Similaridade Semântica no Contexto Biológico e Biomédico

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Edgard Costa Oliveira

Coorientador: Prof.ª Dr.ª Milene Serrano

Brasília, DF

2015

Fabiana Mitsu Alvarenga Ofugi

Busca de Textos Utilizando Similaridade Semântica no Contexto Biológico e Biomédico/ Fabiana Mitsu Alvarenga Ofugi. – Brasília, DF, 2015-

66 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Edgard Costa Oliveira

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. Similaridade Semântica. 2. Ontologias. I. Prof. Dr. Edgard Costa Oliveira. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Busca de Textos Utilizando Similaridade Semântica no Contexto Biológico e Biomédico

CDU 02:141:005.6

Fabiana Mitsu Alvarenga Ofugi

Busca de Textos Utilizando Similaridade Semântica no Contexto Biológico e Biomédico

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 16 de dezembro de 2015:

Prof. Dr. Edgard Costa Oliveira
Orientador

Prof.ª Dr.ª Milene Serrano
Coorientadora

Prof. Dr. Edson Alves da Costa Junior
Convidado 1

Prof. Dr. Sérgio Antônio A. de Freitas
Convidado 2

Brasília, DF
2015

Agradecimentos

Agradeço aos meus pais que me deram o apoio necessário para que eu chegasse até aqui.

Agradeço aos Profs. Drs. Edgard Costa e Milene Serrano por toda a orientação atenciosa e valiosa durante o desenvolvimento deste trabalho. Ao Prof. Dr. Edson Alves por todas suas aulas e orientação em projetos, com toda a sua maestria. Aos Profs. Drs. Georgios Gkoutos e Robert Hoehndorf pela orientação no projeto durante o intercâmbio na Aberystwyth University, além de terem proposto o tema deste trabalho.

Agradeço aos amigos e colegas que me apoiaram direta ou indiretamente, durante o curso, este trabalho e durante a vida, principalmente a Thaiane Braga, Tomaz Martins, Thais Ziober, Débora Marcolino, Guilherme Monteiro, por terem estado ao meu lado durante importantes momentos.

Agradeço também aos professores não citados aqui, mas que se dedicaram a transmitir seus ensinamentos, conhecimentos e orientações com paciência e sabedoria.

"It is a mistake to think you can solve any major problems just with potatoes."
(Douglas Adams)

Resumo

Com o crescente aumento da literatura biológica e biomédica, se faz necessário o uso de buscas que retornem mais do que a tradicional busca por palavras chave oferece. Este trabalho tem como objetivo estudar um método de busca semântica no contexto biológico e biomédico, além de implementar e avaliar um algoritmo com o intuito de propor melhorias. Um estudo bibliográfico com os conceitos utilizados foi conduzido, seguido da caracterização da demanda por um algoritmo neste contexto. O algoritmo utiliza ontologias, permitindo que uma entrada de busca e artigos em uma base, ambos com termos presentes nas ontologias utilizadas, sejam comparados a fim de encontrar os textos mais similares semanticamente. Além disso utiliza em sua implementação a biblioteca *Semantic Measures Library*. Em uma primeira parte do trabalho, o estudo bibliográfico, a caracterização da demanda e a implementação do algoritmo foram concretizados. Na segunda parte foram abordadas as melhorias e avaliações do algoritmo. Com a implementação obtida até o momento, notou-se que os tempos de execução não estão satisfatórios.

Palavras-chave: similaridade semântica. ontologias. busca de textos.

Abstract

With the growing increase in biological and biomedical literature, there has also been a growing need for search mechanisms that provide better returns than what mere keywords search can produce. This paper studies a semantic search method in the biological and biomedical context, as well as implementing and assessing an algorithm so as to propose improvements. It also conducts a bibliographical study of the concepts used, which is followed by the characterisation of the demand for an algorithm within this context. The algorithm uses ontologies, allowing for the comparison of a search entry and given articles – both containing terms present in the ontologies used – so as to find texts that are most similar semantically. Its implementation also includes the use of the Semantic Measures Library. In this first stage of the paper, the bibliographical study, as well as the characterisation of the demand and the implementation of the algorithm have been concluded. The second part approached the improvements and assessment of the algorithm. With the implementation conducted so far, it has been noted that running times are not satisfactory.

Key-words: semantic similarity. ontologies. text search.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DAG	*Directed Acyclic Graph*
FGA	Faculdade do Gama
OWL	*Web Ontology Language*
PMC	PubMed Central
RDF	*Resource Description Framework*
RDFS	RDF *Schema*
SML	*Semantic Measures Library*
TCC	Trabalho de Conclusão de Curso
URI	*Uniform Resource Identifier*
W3C	*World Wide Web Consortium*

Sumário

Parte I

Introdução

1 Introdução

Este capítulo tem o intuito de introduzir o leitor ao tema, apresentando o contexto dentro do qual este trabalho se encontra, a justificativa da proposta, a questão de pesquisa em torno da qual o trabalho está baseado, os objetivos e por fim, a organização do documento.

A busca de textos como é feita na maioria das vezes, ou seja, por meio de palavras-chave, pode trazer como retorno uma quantidade grande de resultados, porém muitos se mostram irrelevantes (BREITMAN, 2005, págs. 2-3). Dessa forma, a busca semântica se apresenta cada vez mais necessária, devido à sua capacidade de buscar resultados considerando o significado e o contexto da busca. No contexto biomédico e biológico, as informações são representadas em sua maioria de forma subjetiva (PESQUITA et al., 2009), dificultando a busca semântica. Assim, o uso de uma representação objetiva se faz necessário, sendo o caso do uso de ontologias, que são capazes de representar o conhecimento em uma especificação formal e explícita de uma conceitualização compartilhada (GRUBER, 1993).

1.1 Contextualização e descrição do problema

A quantidade de literatura biomédica está crescendo rapidamente, a um passo duplo-exponencial (HUNTER; COHEN, 2006), e buscar artigos com informação sobre determinados fenômenos biológicos e publicações relevantes torna-se, portanto, um desafio. O PubMedCentral[1], por exemplo, que é um arquivo de textos completos de literatura biomédica e de ciências da vida da *National Library of Medicine* do Instituto Nacional de Saúde dos EUA, possui 3,6 milhões de artigos disponíveis. Neste contexto, buscar artigos que contenham um dado termo ou um conjunto de termos exatamente como colocado(s) em uma consulta – i.e. uma *query* de busca – pode nem sempre ser útil. Um indivíduo pode desejar encontrar artigos que contenham conteúdo similar ao especificado em uma *query* de busca, ao invés de uma correspondência exata. Uma busca baseada em similaridade semântica poderia auxiliar as pessoas a encontrar conteúdos semelhantes e compará-los, ou mesmo a pesquisar mais informações sobre determinado assunto de interesse.

No intuito de comparar diferentes conteúdos, eles devem estar representados de forma objetiva, o que nem sempre é o caso no campo biológico. Pela própria natureza da área, o conhecimento não pode ser representado com fórmulas, sendo em sua maioria representado e divulgado em linguagem natural, na forma de publicações científicas, ou

[1] <http://www.ncbi.nlm.nih.gov/pmc/>. Acesso em outubro de 2015

de forma que represente uma entidade estruturalmente e não funcionalmente, como uma estrutura genética, por exemplo. Uma abordagem bem sucedida para organizar a informação no contexto biológico é a aplicação de ontologias, já que elas podem representar esses diferentes conteúdos objetivamente (PESQUITA et al., 2009). Ontologias contém a especificação de vocabulários estruturados que podem conter complexos axiomas que relacionam os termos encontrados em um domínio (DOU; MCDERMOTT; QI, 2005). Vale ressaltar que diversas ontologias foram desenvolvidas no campo da biologia e biomedicina, como por exemplo, a *Gene Ontology* (PESQUITA et al., 2009) e a *Human Disease Ontology* (HOEHNDORF; DUMONTIER; GKOUTOS, 2012), que são largamente utilizadas em diversas aplicações.

Uma das aplicações das ontologias é a recuperação da informação na literatura científica (DOMS; SCHROEDER, 2005). Em particular, a estrutura de grafo provida pelas ontologias torna possível navegar pela literatura científica baseando-se na estrutura da própria ontologia. Artigos são classificados com base na ocorrência de termos de ontologias no texto. Assim, a associação resultante entre um ou mais artigos em análise e os termos de ontologias presentes em uma *query* pode então ser utilizada para recuperar o(s) artigo(s), caso essa associação seja bem sucedida. Nessas análises, os conceitos e a estrutura de grafo das ontologias são utilizados, enquanto os axiomas, que especificam relações e restringem o significado dos termos (GRUBER, 1993), não estão sendo aplicados. Acesso automatizado aos axiomas de ontologia requer o uso de um *reasoner* [2] automatizado, para extrair inferências das ontologias (KUBA, 2012).

1.2 Justificativa

A quantidade de dados produzidos na biologia hoje faz do desenho de estratégias para integração de dados entre bancos de dados, métodos para recuperação de dados e desenvolvimentos de linguagens de *query* e interfaces uma parte importante e central da pesquisa biológica. Um dos objetivos das ontologias é lidar com esses crescentes problemas na biologia e biomedicina e prover meios para integrar dados entre múltiplas bases de dados heterogêneas (HOEHNDORF; DUMONTIER; GKOUTOS, 2012).

A forma como as buscas são realizadas em sua maioria, ou seja, utilizando palavras-chave, nem sempre retorna resultados relevantes. A busca pode retornar resultados em diversos contextos, de acordo com os significados que cada palavra pode possuir. Dessa forma, fica a cargo do usuário realizar interpretações e filtrar somente os resultados pertinentes. Com certas organizações da informação, como as ontologias, as buscas poderão ser processadas automaticamente por um computador (BREITMAN, 2005).

[2] Um *reasoner* é um *software* capaz de inferir consequências lógicas de um conjunto de fatos declarados ou axiomas

Diante do exposto, caso dois conteúdos sejam descritos usando a mesma, ou as mesmas, ontologia(s), então os termos utilizados em suas descrições poderão ser comparados a fim de determinar o quão próximos em significado eles são. Com esse resultado, será possível ainda encontrar um valor numérico, que descreve a similaridade semântica (PESQUITA et al., 2009). Dentro do contexto deste trabalho, os termos que serão comparados são o conjunto de termos na *query* de busca fornecida pelo usuário como entrada e os termos contidos nos artigos, para que a similaridade seja computada e os textos mais similares semanticamente sejam recuperados.

1.3 Questão de Pesquisa

A questão de pesquisa que motiva este trabalho é:

É viável, do ponta de vista técnico computacional, implementar um método por meio de um algoritmo que calcule a similaridade semântica entre uma query de busca informada e uma base de artigos científicos na área de biologia e biomedicina, e retorne um ranking de artigos mais similares semanticamente?

1.4 Objetivos

Os objetivos, geral e específicos, definidos para este trabalho, estão relacionados a seguir.

1.4.1 Objetivo Geral

O objetivo geral do trabalho é estudar, implementar, avaliar e propor melhorias para um método de recuperação de textos baseado em similaridade semântica. Tal método, após computar quão similar a *query* de busca do usuário é com relação aos artigos, deverá retornar uma lista ranqueada dos artigos mais similares encontrados em determinada base.

1.4.2 Objetivos Específicos

1. Realizar estudo bibliográfico sobre o tema de similaridade semântica com ontologias no contexto biológico e biomédico para melhor entendimento dos conceitos e algoritmos e estudar produtos de *software* similares;
2. Caracterizar a demanda por um ambiente e um algoritmo de busca por similaridade semântica e ontologias;
3. Especificar, construir e propor melhorias para um algoritmo que permita buscas de textos utilizando similaridade semântica retornando uma lista ranqueada dos artigos similares encontrados;

4. Conduzir os estudos, o desenvolvimento bem como a avaliação do algoritmo usando como base uma metodologia de pesquisa, sendo ela exploratória, quantitativa e qualitativa, e com cenários de uso.

1.5 Organização do documento

Este documento compreende os seguintes itens:

- Parte I - Introdução:

  Capítulo 1 - Introdução: o presente capítulo visa introduzir o tema do trabalho ao leitor, contextualizando e apresentando a justificativa, a questão de pesquisa e os objetivos, geral e específicos.

  Capítulo 2 - Metodologia: descreve as atividades envolvidas no desenvolvimento da proposta e apresenta o cronograma;

- Parte II - Desenvolvimento:

  Capítulo 3 - Referencial teórico: apresenta revisão bibliográfica, com descrição de conceitos e métodos utilizados no trabalho;

  Capítulo 4 - Caracterização da demanda: expõe a demanda inicial que motivou a construção do algoritmo;

  Capítulo 5 - Descrição do Algoritmo: descreve a presente situação e detalha o algoritmo desenvolvido;

- Parte III - Conclusões:

  Capítulo 6 - Resultados: apresenta os resultados obtidos com o trabalho realizado, mostrando os cenários de uso e melhorias;

  Capítulo 7 - Considerações finais: apresenta as conclusões tiradas do trabalho e passos futuros.

2 Metodologia

Este capítulo visa apresentar as metodologias utilizadas para alcançar cada objetivo específico, expondo as classificações em que a pesquisa se enquadra, os métodos a serem utilizados para o desenvolvimento do trabalho e o cronograma da pesquisa.

2.1 Classificação da pesquisa

Segundo Gil (2002), pesquisas podem ser classificadas quanto aos seus objetivos. Neste sentido, pesquisas exploratórias são aquelas que tem como objetivo propor familiaridade com o tópico a ser estudado, a fim de torná-lo mais explícito e aprimorar ideias. Geralmente se dá por meio de estudo bibliográfico. Também existem outros tipos de pesquisa, como por exemplo, descritiva e explicativa.

Além disso, pesquisas podem ser classificadas quanto à sua abordagem, ou seja, podem ser quantitativas ou qualitativas. São quantitativas quando as informações podem ser traduzidas em números e analisadas, e qualitativas quando as informações são subjetivas e não podem ser quantificadas (TAFNER; SILVA, 2007). Outro tipo de classificação é aquele com relação aos procedimentos técnicos. A pesquisa pode utilizar cenários de uso, que descrevem um contexto e suas condições, e podem ser utilizados na fase inicial de *design* provendo informações adicionais sobre o processo e o sistema, o conhecimento do domínio, assim como para exercitar alguma tecnologia específica. Eles provem um conjunto de problemas reais para ajudar na avaliação e fornecem *feedback* (CARROLL, 1995).

Diante do exposto, a pesquisa para este trabalho será do tipo exploratória quanto aos objetivos; quantitativa e qualitativa quanto à abordagem, pois conterá dados numéricos, como tempos de execução, bem como dados não quantificáveis, como análise de similaridade semântica; e quanto ao procedimento utilizará de cenários de uso, que se adaptará conforme testes e avaliações.

2.2 Desenvolvimento da pesquisa

Cada um dos tópicos a seguir descreve os métodos utilizados para alcançar cada um dos objetivos específicos:

1. **Estudo bibliográfico** - Para realizar o estudo bibliográfico sobre o tema de similaridade semântica com ontologias no contexto biológico e biomédico, foram pesquisados conteúdos já publicados em artigos, livros e outros meios, como periódicos e *websites*.

Os conceitos estudados na pesquisa foram organizados e descritos no capítulo 3 de referencial teórico. Foram pesquisados também produtos similares ao proposto, para que sejam utilizados como referência para a implementação da solução proposta.

2. **Caracterização da demanda** - A demanda por um ambiente e algoritmo de busca que utilize similaridade semântica e ontologias foi caracterizada a partir da descrição do projeto que originou este trabalho, além do estudo do referencial teórico, resultando na definição do escopo.

3. **Especificação, construção e avaliação do algoritmo** - O algoritmo foi especificado e desenvolvido após definição do escopo. O desenvolvimento foi dado por iterações, onde testes e avaliações retroalimentaram cada iteração. Para análise quantitativa, foram realizados testes utilizando ferramentas capazes de mensurar variáveis relacionadas ao desempenho, como tempo de execução. Por outro lado, para análise qualitativa, ou seja, para analisar a similaridade semântica entre a *query* de entrada e o artigos retornados, foi feita, em uma primeira fase, uma análise manual para avaliar se os artigos estão no contexto dos termos da *query*. Além disso, foram propostas melhorias ao algoritmo a partir de análises estáticas do código, identificações de possíveis otimizações e testes para melhorar a qualidade.

O registro dos tópicos 1, 2 e 3 supracitados serão realizados com a escrita do documento referente ao Trabalho de Conclusão de Curso (TCC).

A Figura 1 ilustra as atividades do processo de desenvolvimento do TCC. A gestão e acompanhamento do desenvolvimento deste trabalho como um todo se deu por meio da ferramenta de organização Trello.

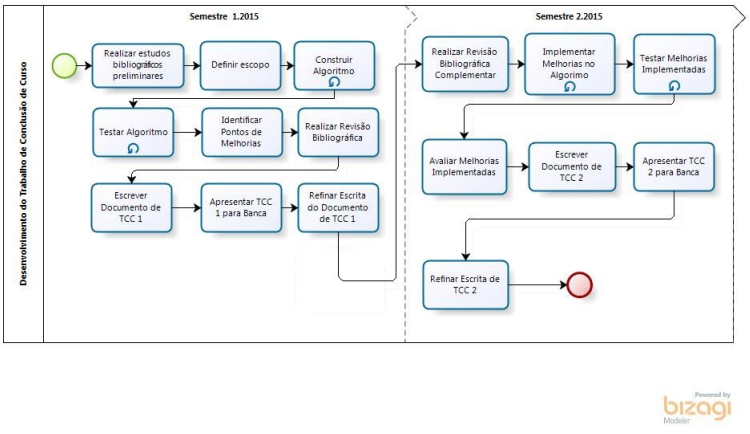

Figura 1 – Processo de desenvolvimento do TCC

Parte II

Desenvolvimento

3 Referencial Teórico

Este capítulo tem como objetivo apresentar o referencial teórico que embasou a pesquisa deste trabalho, incluindo importantes conceitos abordados, tais como a busca semântica, ontologias e a similaridade semântica. Também aborda brevemente um produto similar.

3.1 Busca semântica

As páginas *web* foram um mecanismo desenvolvido por programadores de *software* com o objetivo de compartilhar informações de maneira simples. A informação nessas páginas é exibida através do uso de linguagens de marcação, como HTML ou XML, as quais definem aspectos de exibição e codificação do conteúdo, como tamanho da fonte, cor, posição na tela, *hiperlinks* para outras páginas *web* etc (BREITMAN, 2005). A popularização da internet nas últimas décadas fez com que a quantidade de páginas *web* crescessem de forma exorbitante, e como consequência a recuperação das informações dessas páginas, via mecanismos de busca, tem se tornado cada vez mais complexa (BREITMAN, 2005). Os mecanismos de busca fazem as buscas das páginas *web* de forma sintática, ou seja, levando em conta somente as palavras chaves fornecidas pelo usuário, o que pode trazer resultados inconsistentes devido aos diferentes significados que uma determinada palavra pode assumir, de acordo com o contexto em que é usada. Além disso, as páginas *web* foram construídas de forma que as pessoas pudessem entender, e não para que fossem processadas automaticamente por máquinas. Diante disso, a Web Semântica surgiu como uma nova tendência, com o objetivo de aprimorar a busca de informações na internet atual, adicionando semântica na construção das páginas *web*.

A ideia central da Web Semântica é categorizar a informação de maneira padronizada, como por exemplo em uma taxonomia. Um exemplo de taxonomia é a lineana, que classifica os seres vivos, onde a informação é classificada em hierarquias utilizando generalizações, ou relacionamentos do tipo pai-filho ou tipo-de. A Figura 2 mostra a classificação dos humanos na taxonomia lineana. No caso da Web Semântica, essa classificação seria feita em relação ao conteúdo das páginas *web*, ou seja, o significado semântico daquela página seria escrito no momento da sua construção, utilizando uma linguagem própria que favoreça o processamento dessas informações pelos computadores. Segundo Berners-Lee, Hendler e Lassila (2001), a Web Semântica deve ser o mais descentralizada possível, assim como a *web* atual é. No futuro existirá vários modelos de organização da informação do conteúdo das páginas *web*, onde qualquer empresa, universidade ou organização na *web* poderá ter seu próprio modelo de organização (BERNERS-LEE; HENDLER; LASSILA,

2001).

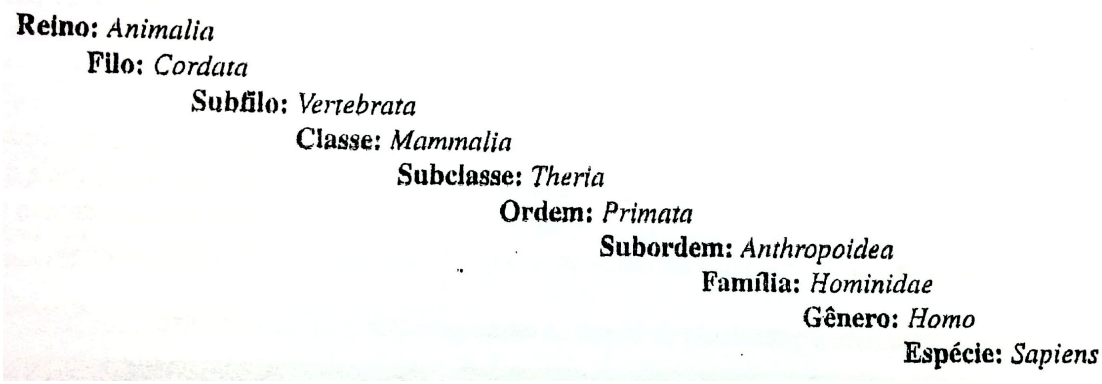

Figura 2 – Classificação dos humanos na taxonomia lineana dos seres vivos (BREITMAN, 2005)

Para que a Web Semântica funcione, os computadores devem ter acesso a coleções estruturadas de informação e conjuntos de regras de inferência que possam usar para conduzir raciocínio automatizado (BERNERS-LEE; HENDLER; LASSILA, 2001). Alguns dos conceitos fundamentais que são pilares dessa nova *web* são os metadados e ontologias, que auxiliam na organização do conhecimento, para que os computadores sejam capazes de processar a informação presente nos documentos ou páginas.

Os metadados servem basicamente para descrever informações sobre objetos ou indivíduos, através da linguagem RDF/RDFS (*Resource Description Framework/Schema*), e são base para a construção de ontologias. As ontologias, por sua vez, servem para dar a semântica necessária por meio do relacionamento entre objetos/indivíduos, relacionamento entre propriedades, domínio de valores de uma propriedade, dentre outros. Nas seções seguintes serão abordados esses conceitos com mais detalhes.

3.2 Metadados

Uma das definições sobre metadados afirma que são "dados sobre dados. O termo se refere a qualquer informação utilizada para a identificação, descrição e localização de recursos" (BREITMAN, 2005, p. 16).

Segundo o W3C, metadados são "informações para a *Web* que podem ser compreendidas por máquinas". Tal definição é mais voltada para a Web Semântica.

Uma das linguagens na qual é possível representar metadados é a RDF (*Resource Description Framework*), descrita a seguir.

3.2.1 RDF - *Resource Description Framework*

As principais funções do RDF são fornecer modelos de dados e sintaxe para a codificação de metadados, de modo que esses possam ser entendidos por máquinas, e

também fornecer a interoperabilidade entre agentes que tenham a necessidade de troca de informações pela internet (BREITMAN, 2005).

Segundo Breitman (2005), o RDF "é uma linguagem declarativa que fornece maneira padronizada de utilizar o XML (*eXtensible Markup Language*) para representar metadados no formato de sentenças sobre propriedades e relacionamentos entre itens na *web*".

O RDF descreve recursos em função de seus relacionamentos e propriedades. Tais recursos podem ser qualquer tipo de item, desde que possuam um endereço único na Internet. Além disso, o RDF garante que cada termo possui uma única definição, através do uso de *namespace* do XML.

Descrições RDF possibilitam que um conteúdo semântico possa ser interpretado por máquinas, principalmente por ser representada no formato XML. Podem ser utilizadas para representar vários tipo de descrição, tais como, itens de compra, informações sobre páginas *web*, conteúdo para máquinas de busca, dentre outros.

O RDF utiliza URIs (*Uniform Resource Identifier*) para identificar recursos, e propriedades para descrever esse recurso. As informações são representadas no formato recurso + propriedade + valor, onde recurso e propriedade devem ter uma URI e valor pode ser uma URI, um valor numérico ou uma cadeia de caracteres. Como exemplo, tem-se a representação da página *web* da autora Karin Breitman:

- Recurso: `http://www.inf.puc-rio.br/~karin/index.html`
- Propriedade: `http://purl.org/dc/elements/1.1/creator`
- Valor: `"Karin Breitman"`

O RDF se tornou um padrão recomendado pelo W3C (*World Wide Web Consortium*) em 2004, ou seja, é também um padrão reconhecido pela indústria e pela comunidade.

3.2.1.1 Elementos do RDF

Existem alguns elementos que são importantes para o RDF em um documento XML. Alguns deles são:

- A versão XML: `<?xml version="1.0"?>`
- A tag `<rdf:RDF>` `</rdf:RDF>`, dentro da qual fica o conteúdo e indica que tal conteúdo é RDF
- O *namespace*, através do elemento `xmlns`. O elemento `xmlns:rdf`, por exemplo, re-

presenta o *namespace* do próprio RDF. Esse elemento deve apontar para o endereço que descreve o vocabulário utilizado naquele *namespace*.

- A descrição do recurso é feita através do elemento `rdf:Description`, e é identificado pelo elemento `rdf:about`

3.2.1.2 RDF Schema

Como o RDF possui número limitado de elementos predefinidos, foi necessária uma ampliacão que permitisse que novas classes e propriedades fossem utilizadas em domínios de comunidades independentes. A solução foi a criação do RDF Schema, que é um *framework* com especificações para a criação de novos Schemas, para definir novas descrições de classes e propriedades. Algumas das classes fornecidas pelo RDF Schema, para que sejam criadas instâncias ou subclasses a partir destas, são (BREITMAN, 2005):

- `rdfs:Resource` – Classe dos recursos
- `rdfs:Class` – Classes das classes
- `rdfs:Literal` – Classe dos literais (ou strings)
- `rfds:Property` – Classe das propriedades
- `rdfs:Statement` – Classe das sentenças reificadas

O RDF Schema também possui classes para definir relacionamentos. Algumas delas são:

- `rdfs:subClassof` – define um relacionamento de herança entre duas classes
- `rdfs:subPropertyof` – define um relacionamento de herança entre duas propriedades
- `rdf:type` – relaciona um recurso a sua classe

Dentre outras classes e propriedades que o RDF Schema fornece, estão o `rdfs:comment`, que permite que comentários sejam associados a um recurso, e o `rdfs:label`, que permite que uma etiqueta seja adicionada a um recurso, ou seja, atribui um nome a um recurso, que pode ser utilizado como nó em um uma representação de grafo. Além destes, um vocabulário também pode ser restringido através do `rdfs:domain`, que especifica qual classe de recursos pode ser o sujeito, ou o item a ser descrito, em uma sentença RDF; e também pelo `rdfs:range`, que especifica a classe de recursos que pode ser o objeto, ou valor, de uma sentença.

3.3 Ontologias

O termo Ontologia teve origem na filosofia, onde significa o estudo da existência das coisas e suas relações e propriedades. Seu objetivo é fornecer sistemas de categorização para organizar a realidade (BREITMAN, 2005).

A mais frequente definição de ontologia ligada à Web Semântica é a proposta por Gruber (1993), onde ele afirma que "ontologia é uma especificação formal e explícita de uma conceitualização compartilhada". Já o W3C define uma ontologia como "a definição dos conceitos e relacionamentos, também referidos como termos, utilizados na descrição e na representação de uma área do conhecimento" (W3C, 2015).

Ontologias podem elevar o funcionamento da *web* de várias formas. Podem ser usadas para melhorar a acurácia de buscas, ou seja, o programa de busca pode procurar somente pelas páginas que se referem a um conceito preciso, ao invés de todos os que usam palavras-chave ambíguas. Aplicações mais avançadas usam ontologias para relacionar informações de uma página com as estruturas de conhecimento e regras de inferências associadas (BERNERS-LEE; HENDLER; LASSILA, 2001). No contexto deste trabalho, a busca utiliza das ontologias para buscar termos precisos dentro de artigos, ao invés de páginas *web*.

Segundo o W3C, ontologias devem prover descrições para os seguintes tipos de conceito:

- Classes;
- Relacionamento entre essas classes;
- Propriedades que essas classes devem possuir.

A Figura 3 exemplifica uma ontologia, mostrando parte de uma ontologia de doenças imunológicas, que descreve algumas classes e propriedades relacionadas a desordens do sistema imunológico.

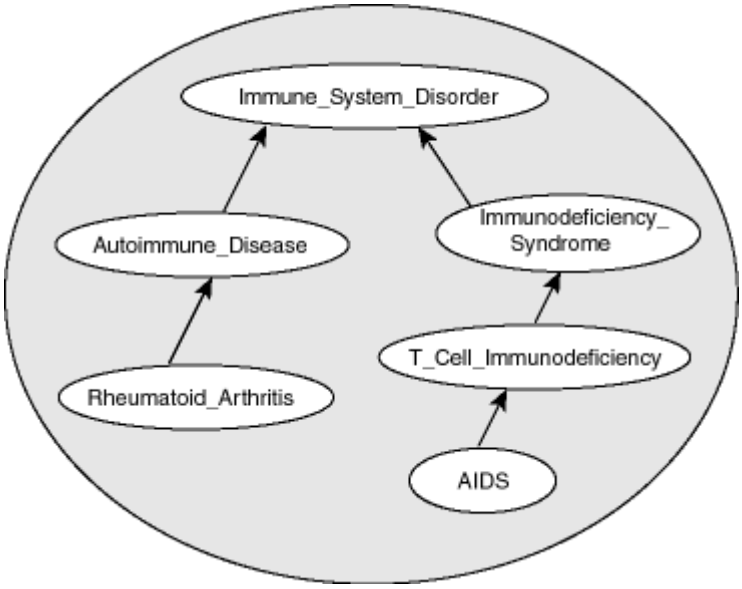

Figura 3 – Exemplo: Ontologia de doenças imunológicas (ORACLE, 2015)

3.3.1 OWL

A *Web Ontology Language* foi lançada pelo W3C e projetada a fim de atender às necessidades de aplicações para a Web Semântica, tais como:

- Construção de ontologias
- Explicitar fatos sobre um domínio
- Racionalizar ontologias e fatos

Segundo Breitman (2005), a linguagem OWL possui alguns elementos básicos. Dentre eles estão:

- *Namespaces*: Para criar um conjunto de conceitos, é necessária a indicação de quais vocabulários estão sendo utilizados. As declarações de *namespaces* são feitas entre etiquetas do tipo `rdf:RDF` permitindo que os identificadores sejam interpretados sem ambiguidades.
- Cabeçalhos: Com os *namespaces* definidos, pode-se incluir uma coleção de sentenças sobre a ontologia com o uso da etiqueta `owl:Ontology`. Estas etiquetas registram comentários, controle de versão, conceitos e informações de outras ontologias.
- Classes: As classes do OWL representam um conjunto ou coleção de indivíduos que compartilham de um grupo de características que os diferenciam dos outros. Classes são utilizadas para descrever conceitos de um domínio, como por exemplo, móveis, animais de estimação, empregados etc.

Em OWL classes são utilizadas para descrever conceitos básicos de um domínio, que vão servir como raízes de diversas taxonomias. Todos os indivíduos pertencem a uma classe genérica `owl:Thing`. Essa conceitualização faz com que toda taxonomia tenha somente uma raiz.

Uma taxonomia tem o construtor `rdfs:subClassOf` que define a hierarquia de classes fazendo uma generalização "tipo-de", como por exemplo, um cachorro é um tipo de animal.

As taxonomias possuem o comportamento transitivo, ou seja, se uma classe Dalmata é uma subclasse de Cachorro, que por sua vez é uma subclasse de Animal, logo Dalmata é uma subclasse de Animal.

- Indivíduos: São objetos do mundo, eles pertencem às classes e se relacionam com outros indivíduos e classes. Em outras palavras, indivíduos são membros das classes.

  Um exemplo da declaração de um indivíduo em OWL poderia ser:
  `<Cachorro rdf:ID="Spike" />`

- Propriedades: Descrevem fatos gerais de uma classe, podendo se referir a todos os membros que pertencem a uma classe, como por exemplo, todos os cachorros comem ração. Podem também se referir a um indivíduo de uma classe, como em: o cachorro Spike nasceu em 2010.

- Restrições: Na linguagem OWL as restrições são feitas utilizando propriedades, e são utilizadas para definir limites para indivíduos pertencentes a uma classe, tais como restrições de cardinalidade ou que utilizam quantificadores, por exemplo.

3.3.2 Ontologias na Biologia e Biomedicina

Na busca por conteúdo significante biológica e clinicamente em meio à atual tecnologia de grande transferência de dados, se fez necessário o uso de formas de relacionar os dados com expressões em vocabulários estruturados e controlados através de anotações, disponibilizando, assim, os dados para busca e processamento por algoritmos. Um dos esforços mais bem sucedidos neste meio foi a *Gene Ontology* (GO), pelo número de usuários e alcance em espécies e granularidade (SMITH et al., 2007).

Em 2001, na tentativa de criar padrões para a criação e desenvolvimento de ontologias de biologia e biomedicina, foi criado o formato de arquivo OBO (*Open Biomedical Ontologies*). OBO é um corpo central para os desenvolvedores de ontologias de ciências da vida, que aplica os conceitos chave por trás do sucesso da GO (SMITH et al., 2007), quais sejam:

- ser abertas, para que estejam disponíveis para uso e sem restrição ou licença, sendo assim aplicáveis em novos propósitos, e também receptivas a modificações através de debates na comunidade;

- ortogonais, para garantir a adição de novas anotações e desenvolvimento modular;
- instanciadas em uma sintaxe bem especificada, para possibilitar o processamento por algoritmos;
- e, por fim, desenhadadas para compartilhar um espaço comum de identificadores, para habilitar compatibilidade retroativa com anotações desenvolvidas anteriormente à medida que as ontologias evoluem.

Devido à necessidade de manter a interoperabilidade entre a OBO e a OWL, que é a linguagem padrão de ontologias, foram criadas diversas ferramentas de conversão que integram os termos de ontologias OBO com os corpos de dados no *framework* da Web Semântica (SMITH et al., 2007).

Posteriormente foi criada a OBO *Foundry*, um projeto colaborativo baseado na aceitação voluntária de seus participantes em desenvolver um conjunto de princípios que extendem os originais da OBO, com os adicionais de que as ontologias sejam desenvolvidas colaborativamente, usem relações comuns definidas de forma não-ambígua, provejam procedimentos para *feedback* do usuário e identificação de versões sucessivas, e possuam limites claros quanto ao conteúdo. O objetivo a longo termo da OBO *Foundry* é que os dados gerados a partir de pesquisa biomédica formem um todo que seja singular, consistente, se expanda cumulativamente e seja tratável algoritmicamente (SMITH et al., 2007).

Existem outras inúmeras aplicações de ontologias biomédicas. De acordo com Hoehndorf, Dumontier e Gkoutos (2012), podem ser utilizadas na análise comparativa de genomas entre múltiplas espécies, e também em análises de fenótipos, ou seja, as características observáveis dos organismos. Além disso, a estrutura de grafo das ontologias biomédicas permite não somente a melhoria de busca e consulta, mas também para tarefas como análise de expressão genética. Análises como esta também são feitas em outros domínios, como o de doenças humanas, com a *Human Disease Ontology*.

A estrutura de grafo das ontologias também é largamente utilizado para análises de similaridade semântica. Outra aplicação das ontologias é em *text mining* e busca e recuperação em literatura. Quando termos de ontologias podem ser detectados em textos em linguagem natural, podem ser utilizadas para recuperação de documentos de texto de arquivos de literatura como o PubMed[1], por exemplo. Assim, se a identificação de termos for combinada com análises, como a de similaridade semântica, a recuperação de textos pode ser melhorada com base na hierarquia da ontologia (HOEHNDORF; DUMONTIER; GKOUTOS, 2012).

Em adição, ontologias também podem ser utilizadas como bases de conhecimento,

[1] <www.ncbi.nlm.nih.gov/pubmed>

para armazenar e expor informação acerca de um domínio. Um outro aspecto de uso das ontologias, além de sua estrutura, é o uso de axiomas, ou seja, afirmações que são consideradas verdadeiras sobre o domínio. As consequências de tais axiomas são inferidas utilizando regras de inferências, que podem ser deduzidas através de raciocínio automatizado (HOEHNDORF; DUMONTIER; GKOUTOS, 2012).

3.4 Similaridade Semântica

Comparação e classificação são princípios importantes quando se trata do estudo de novas coisas dentro da Biologia. Desde Lineu com a sua taxonomia e Darwin com a observação dos beija-flores em Galápagos, comparações são de grande importância ao se estudar novos conceitos, pois pode-se comparar novas entidades com outras conhecidas, fazendo-se inferências sobre suas similaridades. No entanto, no campo biológico, o conhecimento raramente pode ser reduzido a fórmulas matemáticas ou formas objetivas para comparação e é representado na maioria das vezes em linguagem natural, como em publicações, ou utilizando esquemas de classificação (PESQUITA et al., 2009).

Comparar entidades não é uma tarefa trivial. Por exemplo, sequências de genes podem ser comparadas diretamente, mas não quanto aos seus aspectos funcionais. Para que sejam comparados, entidades devem ser representadas de forma objetiva e com propriedades mensuráveis (PESQUITA et al., 2009).

Com o grande aumento de dados a serem processados, se fez necessário o desenvolvimento de maneiras objetivas de representação, para o compartilhamento de conhecimento e processamento computacional. Tais formas de representação são as ontologias, que descrevem termos e suas relações, em sua maioria sendo do tipo “é-um”, que representa uma relação classe-subclasse e do tipo “é parte de”, representando uma relação todo-parte, como pode ser visto na Fig. 4, que mostra uma pequena parte da *Gene Ontology*.

Assim, se duas entidades são anotadas, ou descritas, diretamente ou por herança, com os termos de uma ontologia, utilizando um mesmo esquema, pode-se compará-las comparando seus termos, calculando a similaridade semântica, que é um valor numérico que reflete a proximidade em significado entre essas entidades (PESQUITA et al., 2009).

Uma ontologia estruturada na forma de um grafo acíclico dirigido (*directed acyclic graph* - DAG)[2] permite que a similaridade semântica de termos nessa ontologia possa ser calculada usando duas abordagens: baseada em linhas (*edge-based*) e baseada em nós (*node-based*).

A abordagem baseada em linhas (*edge-based*) consiste principalmente na contagem

[2] Um DAG é um grafo dirigido com nenhum ciclo, ou seja, é formado por uma coleção de vértices e linhas dirigidas, com cada linha conectando um vértice ao outro, de forma que não seja possível começar em um vértice v e seguir uma sequência de linhas que volte para v novamente. (CHRISTOFIDE, 1975)

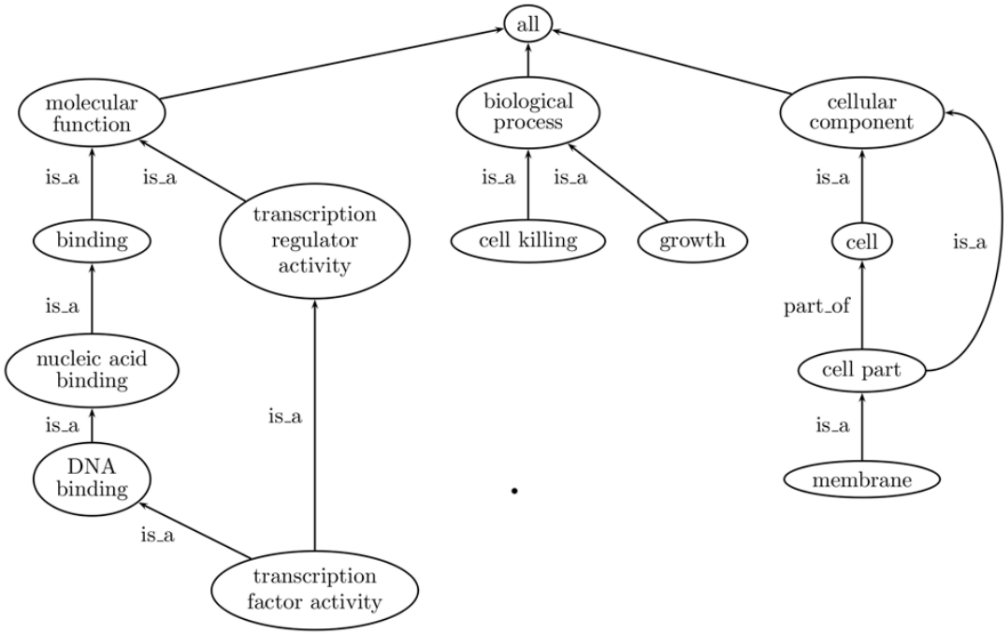

Figura 4 – Exemplo: Trecho da *Gene Ontology* (PESQUITA et al., 2009)

do número de ligações no caminho entre os dois termos sendo comparados. Existem duas técnicas usadas nesta abordagem, a da distância, mais comumente utilizada, e a do caminho comum. A técnica da distância calcula a similaridade semântica através da menor distância entre os dois termos, ou caso exista mais de um caminho entre eles, através da média de todos os caminhos. Já a técnica do caminho comum consiste em calcular a distância entre o ancestral comum mais baixo dos dois termos e o nó raiz. No entanto, raramente em ontologias biológicas os nós e linhas são distribuídos uniformemente e linhas em um mesmo nível correspondem à mesma distância semântica entre termos, o que traz um empecilho para o uso desta técnica em similaridade semântica (PESQUITA et al., 2009).

A outra abordagem, baseada em nós (*node-based*), compara as propriedades entre dois termos, que podem ser relativas aos próprios termos, aos seus ancestrais ou aos seus descendentes. O conteúdo informacional (*information content* - IC) de um termo é uma propriedade comumente utilizada, e diz o quão específico e informativo um termo é, e pode ser calculado com a fórmula:

$$IC = -\log p(c) \tag{3.1}$$

onde $p(c)$ é a probabilidade de ocorrência de c em um corpus[3] específico, geralmente estimado pela frequência de anotação de c no corpus. O conceito do conteúdo informacional pode ser aplicado no ancestral comum entre os dois termos para quantificar a informa-

[3] Segundo o dicionário Merriam-Webster, um corpus é "uma coleção ou corpo de conhecimento ou evidência".

ção que eles compartilham, e então mensurar a similaridade semântica entre eles. Há duas abordagens para se fazer isso: a técnica do ancestral comum mais informativo (*most informative common ancestor* - MICA), onde somente o ancestral comum com maior conteúdo informacional é considerado, e a técnica dos ancestrais comuns disjuntos (*disjoint common ancestors* - DCA), onde todos os ancestrais comuns disjuntos (aqueles que não compartilham de um ancestral em comum) são considerados (PESQUITA et al., 2009). A Figura 5 ilustra o conceito de ancestral comum mais informativo, onde o MICA entre *nickel* e *dime* é *coin*, e o MICA entre *credit card* e *nickel* é o termo *medium of exchange*.

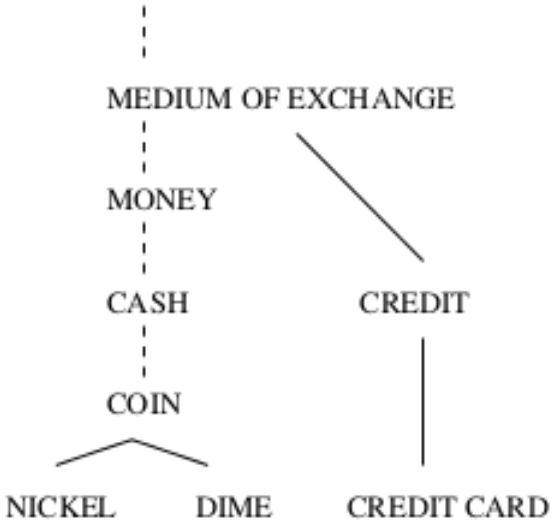

Figura 5 – Fragmento da taxonomia WordNet. (Linhas sólidas representam ligações "é-um" e linhas tracejadas indicam que nós foram omitidos.) (RESNIK, 1995)

Para comparar entidades anotadas com os termos de ontologias, existem duas abordagens, sendo elas por par (*pairwise*) ou por grupo (*groupwise*). A primeira compara termos de uma entidade com termos da outra entidade, sendo que algumas técnicas consideram todas as combinações de pares e outras consideram somente os melhores pares, e depois combinam as similaridades semânticas obtidas a partir dos pares, utilizando comumente a média, máximo ou soma dos valores. A segunda calcula a similaridade considerando não cada termo individualmente, mas um conjunto de termos, grafo ou vetor (PESQUITA et al., 2009).

Existem diversas medidas que calculam a similaridade semântica, uma delas é a de Resnik (RESNIK, 1995), que utiliza simplesmente o conteúdo informacional do ancestral comum mais informativo. Essa medida não leva em consideração a distância do termo até seu ancestral comum. No entanto, outras medidas como a de Lin e a de Jiang e Conrath relaciona o conteúdo informacional do ancestral comum mais informativo com o termo sendo comparado (PESQUITA et al., 2009).

O estudo conduzido por Pesquita et al. (2009) analisa diversas medidas, e conclui que algumas são melhores que outras dependendo da aplicação, principalmente quando

se trata de produtos genéticos e a *Gene Ontology*, que são o foco do estudo. Além disso, explica como escolher a medida de similaridade semântica mais adequada. Entretanto, conclui que se não há necessidade de uma análise detalhada, qualquer das medidas disponíveis é boa o suficiente para dar uma visão geral das similaridades (PESQUITA et al., 2009).

3.5 Produto Similar - GoPubMed

O GoPubMed[4] é um *web server* que faz buscas semânticas na base PubMed[5], da *National Library of Medicine* dos EUA, retornando os resumos de artigos agrupados de acordo com os termos da *Gene Ontology*. O servidor submete a *query* do usuário ao PubMed e identifica termos da ontologia nos resumos, exibindo tais termos destacados. Também destaca os termos utilizados na busca, dando a possibilidade de realizar uma busca por palavras-chave convencional, como pode-se ver na Figura 6.

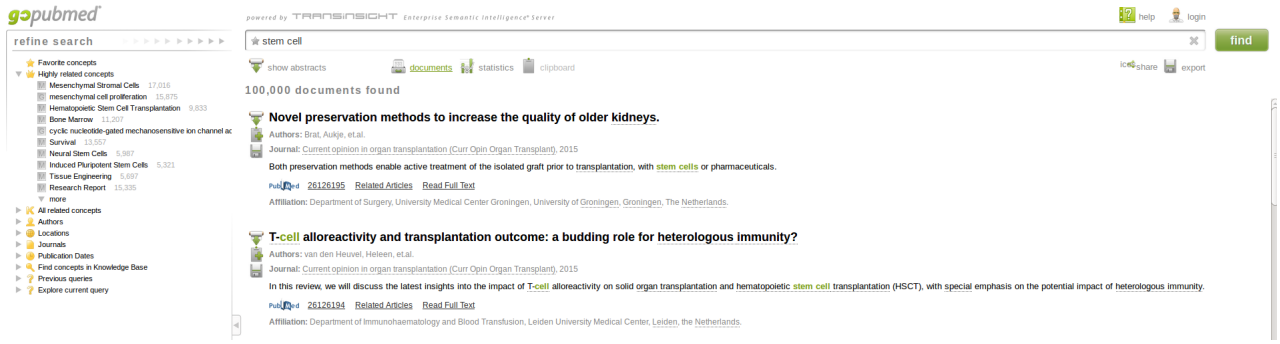

Figura 6 – Busca realizada com a ferramenta GoPubMed

3.6 Resumo do Capítulo

Este capítulo se iniciou explanando o conceito de busca semântica, mostrando a importância de ter uma busca que leva em consideração mais do que as palavras-chave em uma busca, mas também o conteúdo semântico dos termos. Explicou a necessidade em se ter formas objetivas de representar o conhecimento, e detalhou algumas dessas formas, como metadados e a linguagem RDFs para representá-lo, e ontologias com a sua linguagem de representação OWL. O capítulo seguiu mostrando ontologias no contexto biológico e biomédico, com alguns exemplos de ontologias na área e a linguagem OBO, desenvolvida para padronizar a criação de ontologias biológicas. A seguir, explicou o conceito de similaridade semântica, o uso de ontologias e as maneiras de calculá-la. Por fim, descreveu brevemente o GoPubMed, um produto com uma busca similar à proposta.

4 <http://www.gopubmed.org>
5 <http://www.ncbi.nlm.nih.gov/pubmed>

4 Caracterização da demanda

Este capítulo visa dar uma visão geral sobre o algoritmo, além de expor o *background* do mesmo e descrever alguns detalhes técnicos, como o ambiente e ontologias.

4.1 Descrição do projeto antecessor - *Background*

Este projeto teve início em um projeto anterior, realizado na Universidade de Aberystwyth, Reino Unido, como etapa do intercâmbio concluído por meio do programa Ciência sem Fronteiras. Foi realizado no departamento de Ciência da Computacão da universidade com os Profs. Drs. Georgios Gkoutos e Robert Hoehndorf como orientadores, em seu grupo de pesquisa em Biologia Computacional, no período de junho a agosto de 2014. O grupo de pesquisa em Biologia Computacional da Universidade de Aberystwyth conduz pesquisas em áreas que incluem análise de dados biológicos em larga escala, formalização de dados biológicos, informática biomédica, genética, dentre outras. O Prof. Dr. Georgios Gkoutos possui PhD na área de Informática Molecular e Biológica pela Imperial College London e foi pesquisador no departamento de Genética da Universidade de Cambridge, e o Prof. Dr. Robert Hoehndorf possui PhD em Ciência da Computação pela Universidade de Leipzig.

O tema do projeto foi proposto pelos professores e foi dada uma pequena introdução ao domínio no qual trabalho está inserido. Em seguida foram recomendados alguns passos a serem seguidos para alcançar o objetivo do projeto. Foram eles:

1. Utilizar a biblioteca SML (*Semantic Measures Library*) para gerar um grafo a partir de todas as ontologias a serem utilizadas;
2. Adicionar todos os artigos ao grafo como instâncias;
3. Configurar a medida de similaridade - utilizar similaridade *groupwise* e medida de Resnik como medida de CI (conteúdo informacional) para calcular a similaridade;
4. Comparar a *query*, que consiste em um conjunto de IDs de classes de ontologias, com os artigos e encontrar os mais similares;
5. Retornar a lista ranqueada de artigos mais similares;
6. Construir uma interface para realizar a busca.

No capítulo seguinte, sobre o algoritmo, serão descritos quais destes passos foram alcançados dentro do escopo deste trabalho.

4.2 Ambiente

Foram utilizados dois ambientes para rodar o algoritmo, um local e um servidor remoto. O ambiente local possuía as seguintes configurações:

- Sistema Operacional: Ubuntu 14.04.2 LTS 64 bits
- Processador: Intel® Core™ i7-4500U CPU @ 1.80GHz
- Memória RAM: 8GB

O servidor remoto, por sua vez, que era hospedado em uma máquina na Universidade de Cambridge, possuía as seguintes configurações:

- Sistema Operacional: Ubuntu 14.10 64 bits
- Processador: Intel® Xeon® CPU E5-2620 0 @ 2.00GHz
- Memória RAM: 120GB

4.3 Ontologias utilizadas

As ontologias podem ser obtidas do *site* OboFoundry[1], que é um experimento colaborativo envolvendo desenvolvedores de ontologias baseadas em ciência que estão estabelecendo um conjunto de princípios para o desenvolvimento de ontologias, com o objetivo de criar uma suíte de ontologias ortogonais interoperáveis de referência no domínio biomédico (SMITH et al., 2007).

As ontologias utilizadas como entrada para o algoritmo executado localmente, em formato OBO, inicialmente, foram:

- Definições lógicas para termos de regulação multi-espécies

  Estende a ontologia Processo Biológico (*Biological Process*), que provê vocabulários controlados estruturados para a anotação de produtos gênicos com respeito ao seu papel biológico. É um dos três vocabulários da *Gene Ontology*.

  Domínio: Processo Biológico

- Definições lógicas de componentes celulares

  Estende a ontologia Componentes Celulares (*Cellular Components*), que provê vocabulários estruturados controlados para a anotação de produtos gênicos com

[1] <http://www.obofoundry.org/>

respeito às suas localizações celulares. Também é um dos três vocabulários da *Gene Ontology*.

Domínio: Anatomia

- *Chemical Information Ontology*

  Inclui termos para os descritores comumente utilizados em aplicações de *software* de quimioinformática e os algoritmos que os geram.

  Domínio: Bioquímica

- *Gene Ontology - All Logical Definitions*

  Combina a *Gene Ontology*, todas as definições lógicas e os termos de ontologias referenciadas.

- *Human Developmental Anatomy*

  Um vocabulário estruturado controlado de estruturas anatômicas humanas específicas de estágio.

- *Human Phenotype*

  A ontologia de fenótipos humanos (*Human Phenotype Ontology*) está sendo desenvolvida para prover um vocabulário estruturado e controlado para as características fenotípicas encontradas em doenças humanas, hereditárias ou não.

  Domínio: Fenótipo

- *Mosquito Insecticide Resistance*

  Ontologia para entidades relacionadas a resistência a inseticidas em mosquitos.

  Domínio: Ambiente

- *Mouse Pathology*

  Um vocabulário estruturado controlado de fenótipos mutantes e transgênicos de patologias de ratos.

  Domínio: Saúde

Quanto mais ontologias forem carregadas, maior e mais completa será a estrutura utilizada na comparação semântica entre a *query* com termos de ontologias e os artigos, visto que os artigos são instanciados a partir da ligação de seus termos que estão presentes também nas ontologias carregadas, com seu id. No entanto, o tempo de execução aumenta consideravelmente de acordo com a quantidade de ontologias e de artigos. Dessa forma, foram escolhidas poucas e diversas ontologias para fins de testes.

4.4 Resumo do Capítulo

Este capítulo descreveu brevemente o *background* deste projeto, sobre seu início na Universidade de Aberystwyth como parte do intercâmbio realizado pela autora e a demanda inicial. Descreveu também as especificações dos ambientes utilizados para a execução do algoritmo desenvolvido e as ontologias utilizadas para testes locais. O próximo capítulo descreverá o algoritmo em si e alguns resultados parciais.

5 Descrição do algoritmo

Este capítulo fará uma descrição do algoritmo, bem como das ferramentas utilizadas em seu desenvolvimento e alguns resultados parciais obtidos.

5.1 Visão geral do algoritmo

O algoritmo tem como *input* o arquivo que representa a base de artigos anotados, ou seja, representados ou descritos, com termos de ontologias neles contidos; os arquivos das ontologias a serem utilizadas, e a *query*, ou simulação da mesma, para que seja calculada a similaridade semântica com relação aos artigos. Como *output* tem a lista de IDs dos artigos mais similares e suas similaridades. A Figura 7 ilustra as entradas e saídas do algoritmo.

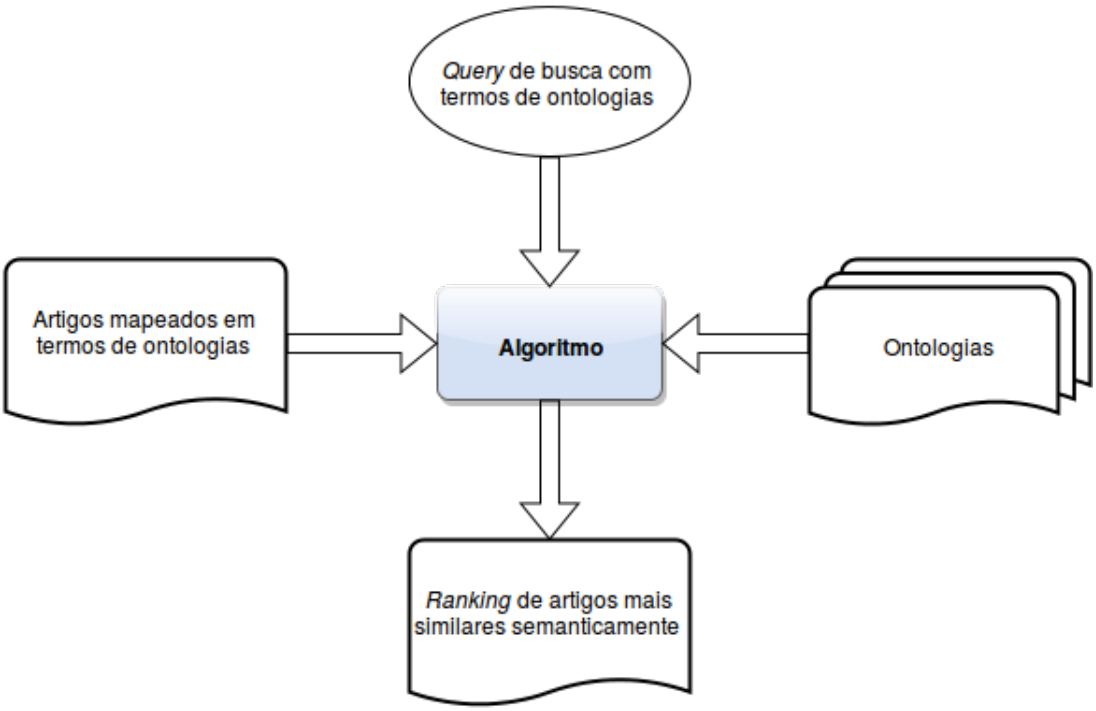

Figura 7 – Entradas e saídas do Algoritmo

Os passos seguidos para a construção do algoritmo foram:

1. Carregar as ontologias de entrada em um grafo, criando uma raiz virtual para que todas as ontologias pertençam ao mesmo grafo, filhas da mesma raiz e remover possíveis instâncias que o grafo já possua, com auxílio da SML.

   As ontologias são lidas de arquivos de um diretório e carregadas em um grafo, e em seguida são reorganizadas abaixo de uma raiz virtual, criada para que todas as

ontologias carregadas estejam no mesmo grafo, filhas dessa mesma raiz. Se houver alguma instância nas ontologias, as mesmas serão removidas. O pseudocódigo a seguir ilustra este passo:

```
// instancia um novo grafo
graph = new GraphMemory(graph_uri);

// carrega as ontologias lidas dos arquivos
for (String ontology : ontologies) {
    GDataConf graphconf = new GDataConf(GFormat.RDF_XML,
    ontologyPath);
    GraphLoaderGeneric.populate(graphconf, graph);
}

// adiciona um vertice correspondente a raiz virtual
graph.addV(virtualRoot);

// faz da raiz virtual a raiz dos grafos de ontologias carregados
GAction rooting = new GAction(GActionType.REROOTING);
GraphActionExecutor.applyAction(rooting, graph);

// remove as instancias
instances = GraphAccessor.getInstances(graph);
graph.removeV(instances);
```

2. Ler o arquivo que contém a representação dos artigos e instanciá-los no grafo. O arquivo contém o ID de cada artigo na base PubMedCentral e termos de ontologias neles presentes.

   Os artigos estão representados da seguinte forma no arquivo de texto:

   ```
   3957068 AAO:0010001 BILA:0000000 CARO:0000000 FBbt_root:00000000
   FMA:62955 HAO:0000000 NIF_GrossAnatomy:birnlex_6 OBI:0100015
   SPD:0000000 TADS:0000583 TGMA:0001822 UBERON:0001062 [...]
   ```

   O primeiro valor, 3957068 neste exemplo, representa o ID do artigo na base PubMedCentral[1] (PMC) onde podem ser recuperados os textos completos, e os demais valores representam termos de ontologias presentes no artigo. Por exemplo, o valor `FMA:62955` é o termo *Anatomical entity*, de ID 62955, da ontologia *Foundational Model of Anatomy*, cujo prefixo é FMA.

   A base PubMedCentral[1] é um arquivo de textos completos e gratuitos de literatura de ciências da vida e biomédicas do *U.S. National Institutes of Health's National Library of Medicine* (NIH/NLM).

[1] <http://www.ncbi.nlm.nih.gov/pmc/>

Os IDs de termos de ontologias (e.g. `FMA:62955`) podem ser consultados no *site* do *Ontology Lookup Service*[2] (OLS), que provê um serviço de interface *web* para pesquisar em múltiplas ontologias com um formato de saída unificado. É um serviço do *European Bioinformatics Institute*.

Os artigos são instanciados criando-se relacionamentos (arestas) do tipo `RDF:type` entre o vértice que representa o ID do artigo e o termo de ontologia que nele ocorre e também está carregado no grafo de ontologias. A Figura 8 ilustra os artigos instanciados no grafo.

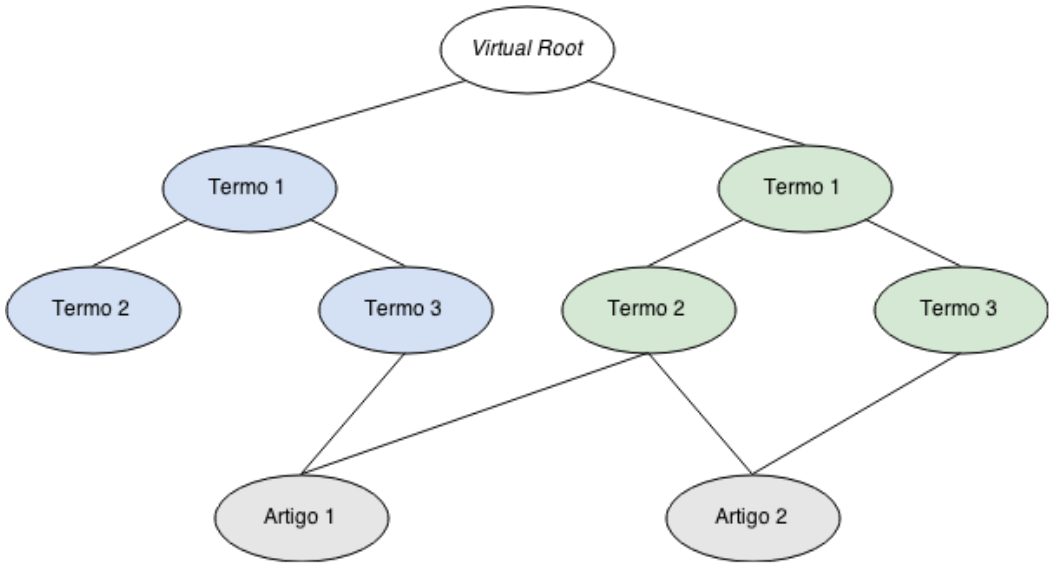

Figura 8 – Exemplo simplificado de artigos instanciados

O pseudocódigo a seguir ilustra o passo de instanciar os artigos no grafo:

```
while (articlesFile.hasNextLine()) {
    line = articlesFile.nextLine();
    splitline = line.split("\t");
    id = splitline[0];
    idUri = factory.getURI(id);
    for (int i = 1; i < splitline.length; i++) {
        if(splitline[i].contains(":")) {
            splitClass = splitline[i].split(":");
            className = splitClass[0];
            classId = splitClass[1];

            classUri = factory.getURI(className + "_" + classId);
            if (graph.containsVertex(classUri)) {
                classUriSet.add(classUri);
            }
        }
    }
```

[2] <https://www.ebi.ac.uk/ontology-lookup/>

```
18      for (URI f : classUriSet) {
19          Edge e = new Edge(idUri, RDF.TYPE, f);
20          graph.addE(e);
21      }
22      if(classUriSet.size() != 0) {
23          articles.put(id, classUriSet);
24      }
25  }
26
```

3. Gerar uma *query* com termos de ontologias para simular uma *query* do usuário.

   A *query* a ser comparada com os artigos deve ser composta por termos de ontologias e os mesmos devem também fazer parte das ontologias que estão carregadas.

4. Calcular a similaridade semântica entre a *query* e os artigos, para obter um *ranking* dos artigos mais similares com relação à *query*. A Figura 9 ilustra o cálculo da similaridade, utilizando os artigos e termos da Figura 8.

   A *query* é então comparada com os artigos a fim de calcular a similaridade semântica, com base na estrutura das ontologias carregadas. O método utilizado foi o Resnik (RESNIK, 1995), com abordagem *groupwise*.

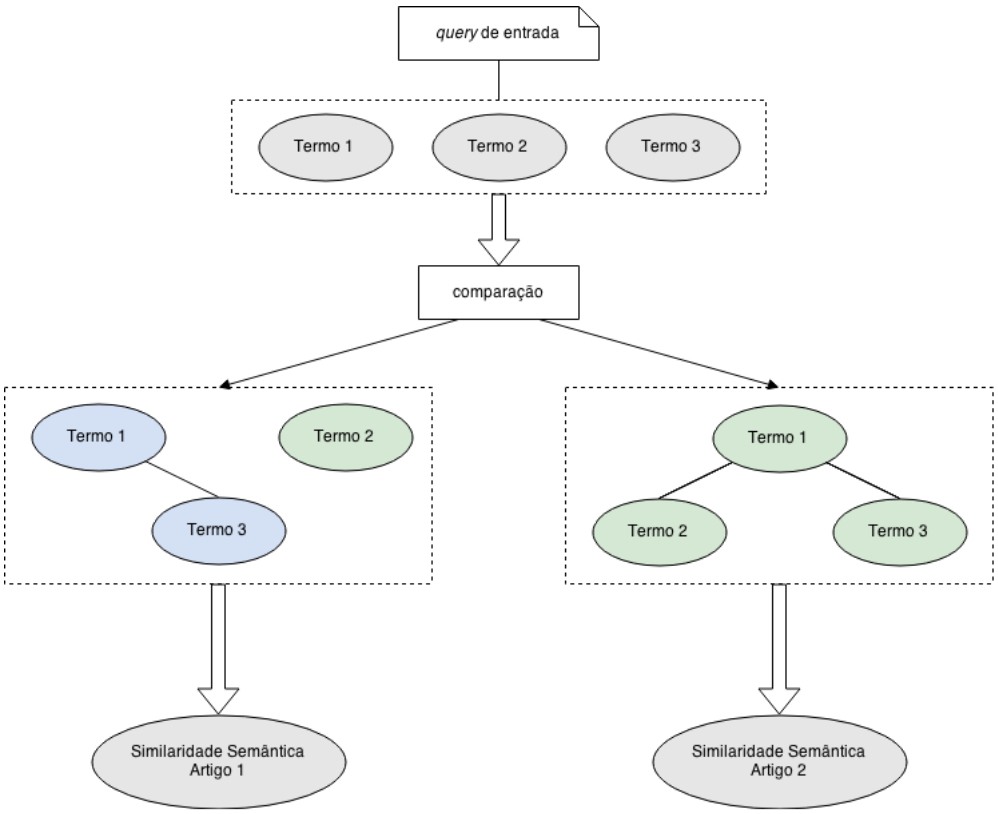

Figura 9 – Exemplo simplificado da similaridade semântica

O pseudocódigo seguinte ilustra o cálculo da similaridade semântica:

```
icConf = new IC_Conf_Corpus("Resnik");

// define a configuracao de medida semantica
smConf = new SMconf("SimGIC");
smConf.setICconf(icConf);

SM_Engine engine = new SM_Engine(graph);

similarityArticles = new TreeMap<Double, String>();
for (Map.Entry<String, Set<URI>> e : articles.entrySet() ) {
    sim = engine.computeGroupwiseSim(smConf, query, e.getValue());
    similarityArticles.put(sim, e.getKey());
}

```

5. Retornar uma lista ranqueada com os IDs dos artigos mais similares e o valor da similaridade semântica encontrada.

5.2 Suporte tecnológico

Esta seção visa listar e descrever as ferramentas e ambiente utilizados para a construção do algoritmo.

5.2.1 Eclipse

Eclipse[3] é uma IDE (*Integrated Development Environment*) *open source* gratuita mais comumente conhecida para o desenvolvimento em linguagem Java. No entanto, pode ser utilizada também com outras linguagens através da instalação de extensões e *plugins*. A versão utilizada inicialmente foi a 4.3, sendo posteriormente atualizada para a 4.5.

5.2.2 Java

Java[4] é uma linguagem de programação orientada a objetos, que roda em todas as plataformas que suportam o Java sem a necessidade de recompilação. As aplicações em Java são tipicamente compiladas pra *bytecode*, que são executados em qualquer JVM (*Java Virtual Machine*), independente de arquitetura ou plataforma, o que torna as aplicações portáveis.

A versão da linguagem utilizada neste trabalho foi a 7. Foi a linguagem escolhida, principalmente, por ser a mesma da biblioteca de apoio SML (*Semantic Measures Library*), que está descrita a seguir.

[3] <https://eclipse.org/ide/>
[4] <https://www.oracle.com/java/index.html>

5.2.3 *Semantic Measures Library* - SML

A SML[5] é uma biblioteca Java genérica e *open source* dedicada à computação e análise de medidas semânticas, como por exemplo, a similaridade semântica e a distância semântica.

Pode ser utilizada para calcular a similaridade semântica entre entidades caracterizadas semanticamente, como conceitos definidos em uma taxonomia ou entidades definidas em um grafo semântico, tais como, documentos, genes ou produtos anotados por conceitos definidos em uma ontologia. Por ser genérica, a biblioteca pode ser utilizada com diversas ontologias e sistemas de organização do conhecimento, como por exemplo, grafos RDF(S), ontologias OWL, WordNet, MesH, ontologias OBO etc.

A versão utilizada neste trabalho foi a SML *Library* 0.8.

5.2.4 Ubuntu

Ubuntu[6] é um sistema operacional Linux gratuito e *open source* baseado no Debian. Seu desenvolvimento é comandado pela Canonical Ltd. e o Projeto Ubuntu é comprometido com os princípios do desenvolvimento de *software open source*, e encoraja as pessoas a utilizarem-no, modificarem-no, melhorarem-no e compartilharem-no, gratuitamente.

A instalação padrão do Ubuntu vem com diversos programas pré instalados que são comumente utilizados pelos usuários, tais como navegadores, cliente de *email*, ferramentas de escritório, dentre outros. Também conta com *firewall* e proteção contra vírus embutidos, além de fornecer atualizações de segurança por até cinco anos na versão LTS (*Long Term Support*).

A versão utilizada no desenvolvimento do algoritmo deste trabalho foi a 14.04 LTS 64-bit.

5.2.5 SonarQube

O SonarQube[7] é uma plataforma *open source* de gerenciamento da qualidade de *software*, anteriormente conhecido apenas como Sonar, dedicada a analisar continuamente e mensurar qualidade técnica, desde projeto a métodos. É possível adicionar extensões com *plugins open source*.

A versão utilizada foi a 5.1.2 para o *server* e 2.4 para o *runner*.

[5] <http://www.semantic-measures-library.org/sml/>
[6] <http://www.ubuntu.com/desktop>
[7] <http://www.sonarqube.org>

5.2.6 EclEmma

O EclEmma[8] é uma ferramenta gratuita de cobertura de código Java para a IDE Eclipse, disponível sob a EPL (*Eclipse Public License*). Traz a análise de cobertura de código diretamente dentro do Eclipse. A versão utilizada foi a 2.3.2.

5.3 Resultados parciais

O algoritmo foi executado localmente com as ontologias descritas no capítulo anterior, uma *query* composta arbitrariamente pelos termos *neural crest* (`EHDA:1355`), *hindbrain* (`EHDA:6488`), *alimentary system* (`EHDA:10251`) e *Global developmental delay* (`HP:0001263`), e 10 artigos.

Os tempos de execução estão na Tabela 1.

Atividade	**Tempo (ms)**
Carregar ontologias	10529
Instanciar artigos	326
Computar similaridade	85845
Total do programa	96949

Tabela 1 – Tempo de execução

Dado que o tempo total deste teste foi superior a 1 minuto para poucas entradas, conclui-se que este não é um tempo razoável para uma busca, devendo então ser submetido a melhorias visando melhor desempenho.

A saída do algoritmo foi a seguinte lista ranqueada, onde o primeiro valor é o ID do artigo, cujo texto pode ser buscado no *site* do PMC, e o segundo valor é a similaridade semântica:

1. `3944782 0.7272727272727273`
2. `2943046 0.6363636363636364`
3. `3816597 0.5454545454545454`
4. `2764347 0.36363636363636365`
5. `3670526 0.2727272727272727`
6. `3851555 0.18181818181818182`
7. `1180857 0.09090909090909091`

[8] <http://eclemma.org/>

O primeiro item é um artigo cujo título é "*The zebrafish anatomy and stage ontologies: representing the anatomy and development of* Danio rerio" que descreve o desenvolvimento de uma ontologia sobre o peixe-zebra, que leva em consideração várias fases de seu desenvolvimento, incluindo seu desenvolvimento neural.

O segundo item possui o título "*Progressive logopenic/phonological aphasia: Erosion of the language network*", que discorre sobre afasias progressivas, que são desordens de linguagem associadas degeneração cerebral.

Pode-se concluir, então, que os itens retornados de fato estão no contexto dos termos da *query*, mesmo com a quantidade limitada de entradas.

5.4 Resumo do capítulo

Este capítulo detalhou a construção do algoritmo, quais entradas são esperadas, ou seja, as ontologias, uma *query* com termos de ontologias e artigos anotados também com termos de ontologias, e qual a saída retornada, i.e. uma lista ranqueada de artigos mais similares encontrados com relação à *query*. Foram apresentadas também as ferramentas utilizadas em seu desenvolvimento, bem como a linguagem e a biblioteca. Por fim, descreveu brevemente uma das execuções do algoritmo localmente, apresentando seus resultados e tempos de execução, que não foram considerados satisfatórios. A saída com os artigos e suas similaridades, por sua vez, foi considerada razoável, visto que os artigos estão no contexto da *query* de entrada.

Parte III

Conclusões

6 Resultados

Este capítulo tem como objetivo expor os resultados obtidos a partir da especificação, implementação e iterações de melhorias do algoritmo proposto. Os ciclos de melhorias serão mostrados na forma de cenários, onde cada um irá expor o estado e as condições em que o algoritmo foi executado, juntamente com as mudanças realizadas em cada iteração.

6.1 Cenário 1

Este cenário expõe o algoritmo apresentado no Capítulo 5, no mesmo ambiente e mesmas condições, detalhando-o, apresentando análises e melhorias realizadas nesta iteração.

O algoritmo citado possuía uma classe de maior importância além da classe principal (Main), denominada OntologiesGraph, ilustrada na figura 10.

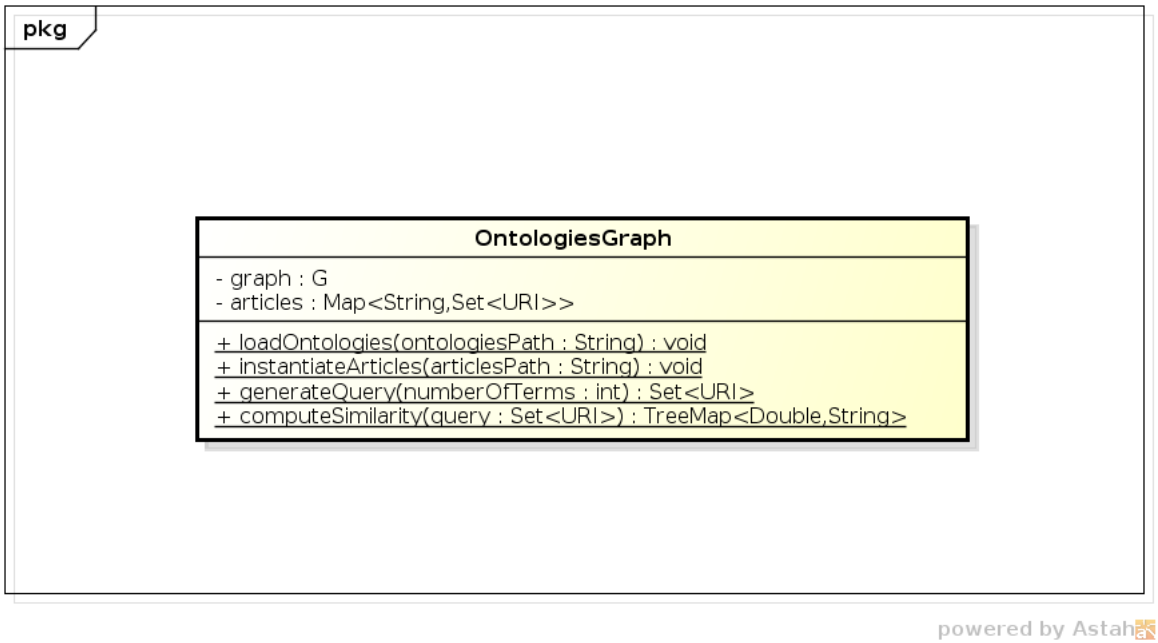

Figura 10 – Classe OntologiesGraph

Os métodos implementados na classe executavam as principais ações necessárias para o carregamento das ontologias a partir de arquivos, para a instanciação dos termos contidos em artigos mapeados e para a comparação de uma *query* com os termos das ontologias carregadas. Tais métodos estão descritos nos itens a seguir:

- **void loadOntologies(String ontologiesPath)** - método estático que carrega as ontologias em um grafo a partir dos arquivos que as contém, em sua maioria com

extensão .obo. É um método sem retorno que recebe como parâmetro o caminho para o diretório onde se encontram os arquivos de ontologias. Itera sobre o diretório populando o grafo com cada uma das ontologias nele presente. Posteriormente, uma nova raiz virtual é criada para que as ontologias carregadas sejam filhas da mesma raiz.

- **void instantiateArticles(String articlesPath)** - método estático utilizado para instanciar os termos de ontologias contidos em artigos, provenientes do arquivo indicado pelo caminho recebido como parâmetro de entrada, no grafo. Ou seja, criar uma aresta (relacionamento) do tipo `RDF:type` entre um nó que representa o artigo e cada um dos termos presentes no artigo e no grafo de ontologias.

- **Set<URI> generateQuery(int numberOfTerms)** - método estático utilitário para gerar uma *query* com termos para simular uma *query* de entrada do usuário. Recebe como parâmetro o número de termos que a *query* irá conter e retorna um conjunto de termos de ontologia.

- **TreeMap<Double, String> computeSimilarity(Set<URI> query)** - método estático que calcula a similaridade semântica entre a *query*, recebida como parâmetro de entrada, e os artigos instanciados. Retorna um objeto do tipo TreeMap, com o valor de similaridade semântica como chave e o id do artigo como valor, para que possam ser ordenados de formas descendente.

Após análise estática utilizando a ferramenta SonarQube, foram coletados alguns dados sobre o código, como se pode ver na Figura 11.

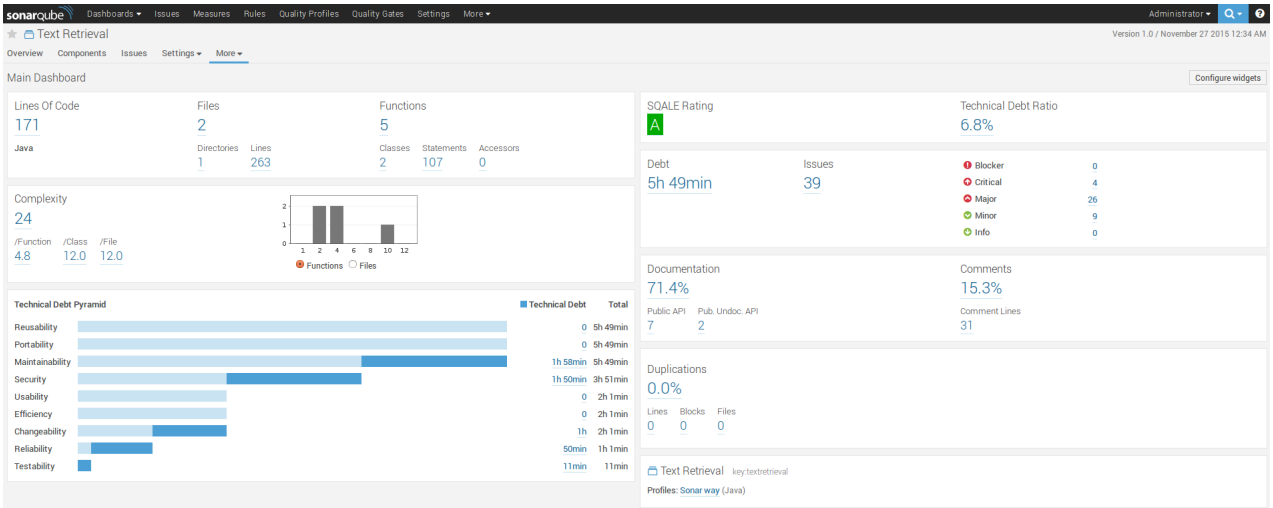

Figura 11 – Análise pela ferramenta SonarQube

O SonarQube utiliza o conceito de Débito Técnico (*Technical Debt*) [1], que estima o tempo que seria gasto para resolver *issues* encontradas relacionadas a requisitos não-funcionais, tais como testabilidade, confiabilidade, usabilidade, manutenibilidade, dentre outros. A proporção de Débito Técnico (*Technical Debt Ratio*) mostra a relação entre o débito técnico total e o tamanho do programa. Ou seja, uma mesma quantidade de débito técnico em um programa maior ou menor, terá menor proporção no programa maior. Uma classificação de acordo com a metodologia SQALE (*Software Quality Assessment based on Lifecycle Expectations*) é dada, sendo A para melhor classificação e E para pior. A complexidade é definida de acordo com a complexidade ciclomática do código, que varia de acordo com a número de caminhos que o código pode seguir (MCCABE, 1976).

A análise encontrou 39 *issues*, algumas consideradas críticas, ou de maior ou menor importância. Dentre elas estavam:

- Utilizar construtor privado ao invés de construtor público em classe utilitária
- Utilizar *logs* ao invés de imprimir mensagens com System.Err ou System.out
- Logar ou relançar exceções
- Reduzir complexidade ciclomática no método **instantiateArticles**
- Refatorar para não deixar mais do que três estruturas de repetição ou de controle (if/for/while/switch/try) aninhadas

A Figura 12 mostra parte das *issues* geradas e sua classificação.

O SonarQube também mostra a porcentagem de documentção em APIs públicas, neste caso resultando em 71,4%, caracterizando a maior parte das APIs como documentadas.

A pirâmide de Débito Técnico indica a quantidade de esforço devida em cada um dos requisitos não funcionais, cumulativamente de baixo para cima.

6.2 Cenário 2

Após resolver a maioria das *issues* encontradas pelo SonarQube, o resultado da análise foi o descrito na Figura 13.

Pode-se notar que tanto a quantidade de *issues* reportadas como o Débito Técnico reduziram de forma significante, restando apenas 6 *issues*, listadas na Figura 14.

A primeira delas refere-se a criar um construtor privado para não deixar o construtor público padrão disponível para instanciar uma classe cujos membros são estáticos.

[1] <http://docs.sonarqube.org/display/SONAR/Technical+Debt>. Acesso em dezembro de 2015

Figura 12 – Algumas *issues* geradas pela ferramenta SonarQube

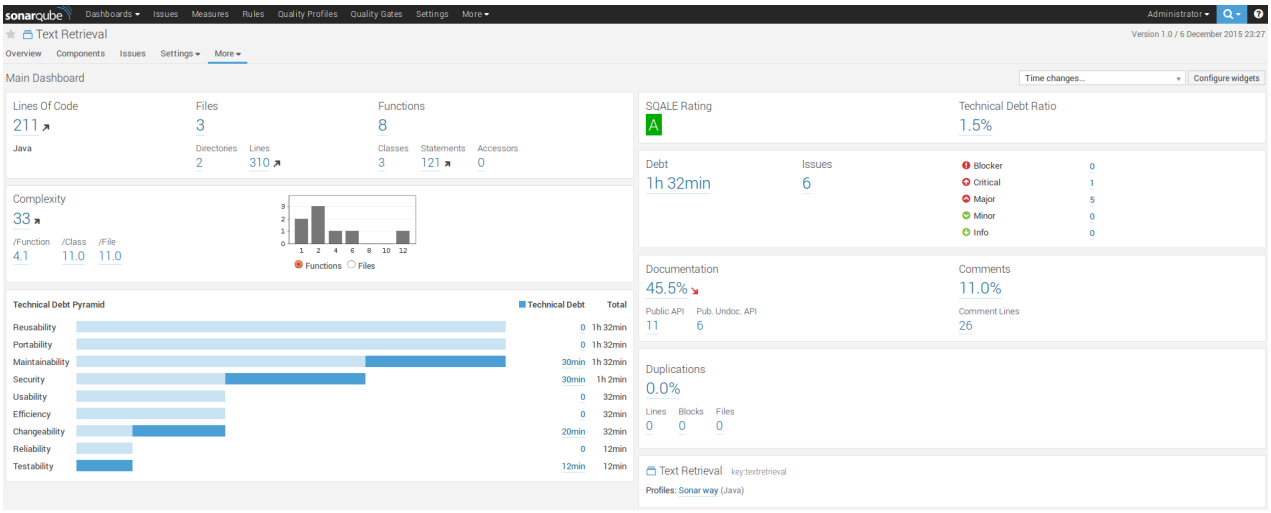

Figura 13 – Análise pela ferramenta SonarQube após melhorias

Nesse caso, trata-se da classe Main, não sendo necessária a criação de um construtor privado. A segunda *issue*, por sua vez, sugere o uso de um *logger* ao invés da saída de sistema System.out. No entanto, optou-se por manter dessa forma para melhor visualização da saída.

As duas *issues* seguintes, referentes à classe OntologiesGraph, tratam sobre um atributo público estático, que anteriormente era privado, porém torná-lo público foi uma alternativa para que fosse viável testar o método. Por fim, as últimas *issues* dizem respeito à complexidade do método **instatiateArticles**. Este método tem alta complexidade devido à grande quantidade de laços de repetição e estruturas condicionais aninhadas. Por ser um método que faz *parse* de um arquivo de texto, tais estruturas são esperadas.

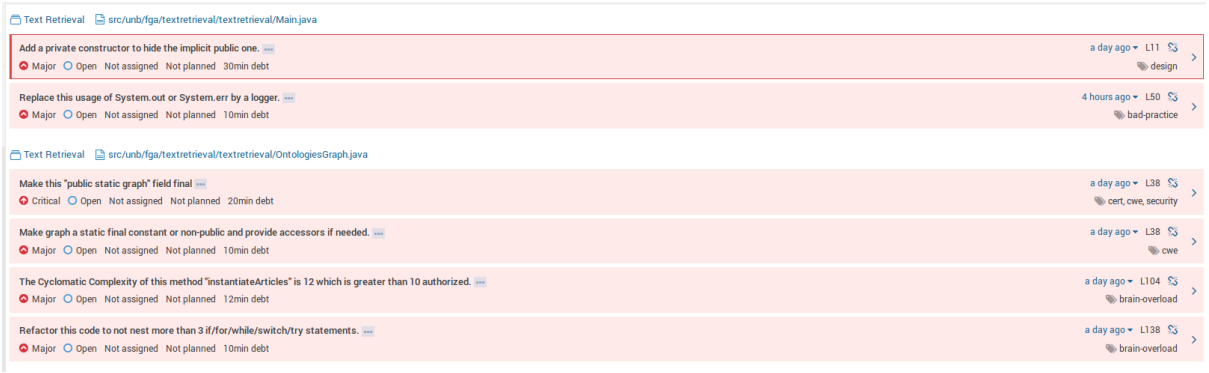

Figura 14 – *Issues* geradas pela ferramenta SonarQube após melhorias

Além disso, no cenário anterior não havia testes unitários e, portanto, nenhuma cobertura de código. Foram implementados testes unitários que garantiram a cobertura de 90% do código da classe OntologiesGraph, como pode ser visto na Figura 15, considerando que a classe Main apenas faz chamadas a esta classe e imprime o resultado.

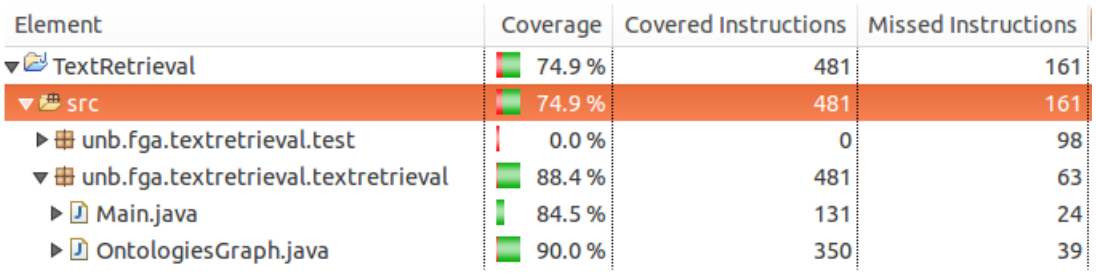

Figura 15 – Cobertura de código pela ferramente EclEmma

Os testes validaram o carregamento das ontologias ao verificar se o grafo estava populado após a chamada do método, se o método lançaria uma exceção caso houvesse algum erro no carregamento, se os artigos foram instanciados corretamente e se o método lançaria uma exceção se não houvesse o arquivo especificado.

Após realizadas as mudanças no código, os tempos de execução passaram a ser os exibidos na Tabela 2

Atividade	**Tempo (ms)**
Carregar ontologias	8175
Instanciar artigos	100
Computar similaridade	79871
Total do programa	88149

Tabela 2 – Tempo de execução após melhorias

Os resultados de similaridade permaneceram os mesmos do cenário anterior.

7 Considerações Finais

Com a revisão bibliográfica, foram identificados e caracterizados diversos conceitos importantes utilizados ao longo do trabalho, como a busca semântica e suas vantagens sobre a tradicional busca por palavras-chave e as formas de organização do conhecimento utilizadas para auxiliar na busca semântica, tais como a linguagem RDF e as ontologias. Também foi descrita a técnica da similaridade semântica e seu uso no contexto biológico e biomédico, e também para recuperar textos em bases a partir desta medida.

Foi apresentado o projeto que deu origem a este trabalho, além de exposta a importância da busca semântica no contexto biológico e biomédico, visto que atualmente a quantidade de artigos disponíveis para pesquisa cresce cada vez mais. O tema de busca utilizando similaridade semântica e ontologias também serviu para reforçar a importância da Web Semântica, tópico cada vez em alta no contexto computacional. Parte do tema deste trabalho foi apresentado como projeto da disciplina de Web Semântica da FGA (Faculdade do Gama), no 1º semestre de 2015. A busca semântica pode não somente ser útil no campo biológico e biomédico, mas também em outras áreas que tenham ontologias desenvolvidas e consolidadas, para buscar em bases de artigos, publicações, ou qualquer repositório de textos que possam ser representados utilizando termos, ou classes, de ontologias.

O algoritmo foi descrito, assim como as ferramentas utilizadas para que seu desenvolvimento fosse possível. Foram feitas análises estáticas do código implementado utilizando a ferramenta SonarQube, para identificar possíveis melhorias a serem feitas. A partir das *issues* encontradas pela ferramenta, realizou-se melhorias no código, a fim de obter maior qualidade na implementação, levando em consideração boas práticas de programação. Ademais, foram adicionados testes unitários, para garantir o bom funcionamento dos métodos implementados. Até então, existe uma versão do algoritmo desenvolvido, que retorna um *ranking* de artigos mais similares à *query*, porém com tempos de execução ainda muito altos.

Pelo uso de uma biblioteca externa, há limitações de *performance*. Uma delas, conhecida pelos criadores da biblioteca e reportada no *site*, diz respeito ao carregamento das ontologias. A base de conhecimento, i.e. as ontologias, são carregadas em memória, permitindo acesso rápido às informações necessárias para calcular medidas semânticas. No entanto, para processar grafos muito grandes, compostos de centenas de milhões de relacionamentos, isto pode ser um problema, o que é o caso de quando se carrega muitas ontologias que contém muitas classes e relacionamentos. Este é um grande limitante quando se trabalha com ontologias grandes, caso recorrente na área biológica e biomédica, e quando se tem como entrada um grande número de representações de artigos, o

que adiciona ainda mais relacionamentos ao grafo.

Assim, pode-se concluir que é viável a implementação de um algoritmo que retorne artigos mais similares semanticamente a partir de uma base, entretanto, são necessárias melhoras em termos de *performance*, para que seja utilizável do ponto de vista do usuário.

Para trabalhos futuros, além do escopo deste trabalho, sugere-se que seja implementada uma ferramenta, *web* ou não, que permita que o usuário insira a sua própria *query* para busca e obtenha os resultados de artigos mais similares. Além disso, sugere-se submeter os resultados da busca a avaliações de especialistas nas áreas biológica e biomédica para validar a eficácia no quesito de similaridade. Vale ressaltar também que a biblioteca *Semantic Measures Library* é *open source*, sendo possível tentar encontrar meios de melhorar a *performance*.

Referências

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific American*, May 2001. Citado 3 vezes nas páginas 31, 32 e 35.

BREITMAN, K. *Web Semântica: a Internet do Futuro*. Rio de Janeiro, Brasil: LTC, 2005. Citado 9 vezes nas páginas 13, 23, 24, 31, 32, 33, 34, 35 e 36.

CARROLL, J. M. *Scenario-Based Design: envisioning work and technology in system development*. New York: John Wiley & Sons, 1995. Citado na página 27.

CHRISTOFIDE, N. *Graph Theory: An Algorithmic Approach*. [S.l.]: Academic Press, Incorporated, 1975. Citado na página 39.

DOMS, A.; SCHROEDER, M. Gopubmed: exploring pubmed with the gene ontology. TU Dresden, 2005. Citado na página 24.

DOU, D.; MCDERMOTT, D.; QI, P. Ontology translation by ontology merging and automated reasoning. In: *Ontologies for Agents: Theory and Experiences*. [S.l.]: Birkhäuser Basel, 2005. p. 73–94. Citado na página 24.

GIL, A. C. *Como Elaborar Projetos de Pesquisa*. São Paulo: Editora Atlas, 2002. Citado na página 27.

GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, v. 5, n. 2, p. 199–220, 1993. Citado 3 vezes nas páginas 23, 24 e 35.

HOEHNDORF, R.; DUMONTIER, M.; GKOUTOS, G. V. Evaluation of research in biomedical ontologies. *Briefings in Bioinformatics*, v. 14, n. 6, p. 696–712, 2012. Citado 3 vezes nas páginas 24, 38 e 39.

HUNTER, L.; COHEN, K. B. Biomedical language processing: Perspective what's beyond pubmed? University of Colorado, 2006. Citado na página 23.

KUBA, M. Owl 2 and swrl tutorial. 2012. Disponível em: <http://dior.ics.muni.cz/~makub/owl/#ontology>. Citado na página 24.

MCCABE, T. J. A complexity measure. *IEEE Transactions on Software Engineering*, v. 2, n. 4, December 1976. Citado na página 59.

ORACLE. Database semantic technologies developer's guide. 2015. Disponível em: <http://docs.oracle.com/cd/B28359_01/appdev.111/b28397/owl_concepts.htm#RDFRM200>. Citado 2 vezes nas páginas 13 e 36.

PESQUITA, C. et al. Semantic similarity in biomedical ontologies. *PLoS Computational Biology*, v. 5, n. 7, jul. 2009. Citado 8 vezes nas páginas 13, 23, 24, 25, 39, 40, 41 e 42.

RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. Massachusetts, EUA, 1995. Citado 3 vezes nas páginas 13, 41 e 50.

SMITH, B. et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, v. 25, n. 11, p. 1251–1255, 2007. Citado 3 vezes nas páginas 37, 38 e 44.

TAFNER, E. P.; SILVA, R. *Apostila de Metodologia Científica*. Associação Educacional do Vale do Itajaí-Mirim, 2007. Citado na página 27.

W3C. Ontologies. 2015. Disponível em: <http://www.w3.org/standards/semanticweb/ontology>. Citado na página 35.